计算机科学
COMPUTER SCIENCE

基于隐私保护的反向传播神经网络学习算法

王健

引用本文

王健. 基于隐私保护的反向传播神经网络学习算法[J]. 计算机科学, 2022, 49(6A): 575-580.

WANG Jian. Back-propagation Neural Network Learning Algorithm Based on Privacy Preserving[J]. Computer Science, 2022, 49(6A): 575-580.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.211100155

基于隐私保护的反向传播神经网络学习算法

王 健
河南财经政法大学计算机与信息工程学院 郑州 450000

摘 要 反向传播神经网络学习算法已经被广泛地应用在医疗诊断、生物信息学、入侵检测、国土安全等领域。这些应用领域的共同点是,都需要从大量的复杂的数据中抽取模式和预测趋势。在以上这些应用领域中,如何保护敏感数据和个人隐私信息是一个重要的问题。目前已有的反向传播神经网络学习算法,绝大多数都没有考虑在学习过程中如何保护数据的隐私信息。文中为反向传播神经网络提出基于隐私保护的算法,适用于数据被水平分割的情况。在建造神经网络的过程中,需要为训练样本集计算网络权向量。为了保证神经网络学习模型的隐私信息不被泄露,本文提出将权向量分配给所有参与方,使得每个参与方都具有权向量的一部分私有值。在对各层的神经元进行计算时,使用安全多方计算协议,从而保证神经网络权向量的中间值和最终值都是安全的。最后,被建造好的学习模型被所有参与方安全地共享,并且每个参与方可以使用该模型为各自的目标数据预测出相应的输出结果。实验结果表明,所提算法在执行时间和准确度误差上比传统非隐私保护算法更具优越性。

关键词:神经网络;隐私保护;安全多方计算;隐私泄露
中图法分类号 TP183

Back-propagation Neural Network Learning Algorithm Based on Privacy Preserving

WANG Jian
College of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou 450000, China

Abstract Back-propagation neural network learning algorithms based on privacy preserving are widely used in medical diagnosis, bioinformatics, intrusion detection, homeland security and other fields. The common of these applications is that all of them need to extract patterns and predict trends from a large number of complex data. In these applications, how to protect the privacy of sensitive data and personal information from disclosure is an important issue. At present, the vast majority of existing back-propagation neural network learning algorithms don't consider how to protect the data privacy in the process of learning. This paper proposes a back propagation neural network algorithm based on privacy-preserving, which is suitable for horizontally partitioned data. In the construction process of neural networks, it is need to compute network weight vector for the training sample set. To ensure the private information of neural network learning model can not be leaked, the weight vector will be assigned to all participants, so that each participant owns a part of private values of weight vector. In the calculation of neurons, we use secure multiparty computation, thus ensuring the middle and final values of the neural network weight vector are secure and will not be leaked. Finally, the constructed learning model will be securely shared by all participants, and each participant can use the model to predict the corresponding output for their respective target data. Experimental results show that the proposed algorithm has advantages over the traditional non-privacy protection algorithm in execution time and accuracy error.

Keywords Neural network, Privacy preserving, Secure multiparty computation, Privacy leakage

1 引言

在机器学习和数据挖掘中,可以使用神经网络学习算法从大量的复杂的数据集合中抽取信息和预测趋势。虽然经过学术界的努力,目前已提出了多种不同的神经网络学习算法,但是这些算法中大多数都没有考虑神经网络学习过程中如何避免隐私信息泄露的问题[1]。

神经网络有多种不同的算法来生成各种不同的模型,例如自组织映射网络、反向传播神经网络、感知学习网络、Hopfield网络等[2]。在这些神经网络模型中,反向传播神经网络由于具有自组织、自适应、容错性和非线性等特点,使其成为机器学习领域的前沿技术,并且在模式识别、联想记忆、函数逼近、复杂控制、信号处理等领域得到广泛应用[3]。基于上述优点,反向传播神经网络学习算法已成为神经网络应用中使用范围最广的算法。因此本文选择为反向传播神经网络学习算法设计基于隐私保护的算法,从而保证神经网络学习过程中的隐私信息不被泄露。

目前研究基于隐私保护的神经网络学习算法的文章较少,可查阅到的文献资料主要有文献[4-6]等。在文献[4]中,Barni提出了一种基于隐私保护的神经网络学习协议。该

基金项目:河南省科技厅科技攻关项目(222102210289)
This work was supported by the Science and Technology Research Project of Henan Provincial Department of Science and Technology (222102210289).
通信作者:王健(goodjian121@126.com)

协议涉及到两个参与方，一方是客户端或数据拥有者，负责输入数据；另一方是服务器或神经网络拥有者，负责处理数据。在整个学习过程中，双方都希望自己的信息不被泄露。该协议是假设神经网络学习模型已经存在，并且可以被用来处理输入数据和预测相应的输出。Barni 针对不同程度的安全考虑提出了 3 种不同的算法。在第一种算法中，两个参与方使用安全点积协议来计算网络中权值的和，并且权值由服务器提供。在第二种算法中，激活函数是私有的，并且只有服务器知道该激活函数。该算法使用 OPE(Oblivious Polynomial Evaluation)[7] 子协议进行私有函数值的安全计算，以防止激活函数的信息被泄露给客户端。在第三种算法中，服务器为了防止客户端准确地预测出神经网络模型，在保证最终输出结果不变的情况下，服务器会向系统添加一些虚假的神经元，并且重新设置一些边界权值。

Barni 提出的上述 3 种算法存在以下安全威胁：首先，神经网络模型拥有者(即服务器)的隐私信息会受到威胁，这是因为数据拥有者即客户端在向服务器发送一系列请求后，就能够很容易地识别出神经网络学习模型；其次，虽然第二种算法中的 OPE 子协议被用来隐藏激活函数，以防止客户端知道此激活函数，但是当客户端接收到服务器返回给它的请求结果后，这个激活函数就可能会被泄露给客户端。

本文提出的基于隐私保护的神经网络学习算法不同于文献[4]中的方法。因为文献[4]中是假设神经网络学习模型已经存在，并且由服务器拥有该网络模型。然而在本文提出的算法中，各个参与方通过使用安全多方计算子协议，安全地生成神经网络学习模型；另外，在本文算法中，敏感性攻击不能把某一方的隐私信息泄露给其它参与方，因为该算法没有使用客户端-服务器的方法，也没有使用 OPE 子协议；此外，文献[4]提出的方法只适用于两个参与方，而本文提出的算法可以适用于多个参与方。

文献[5]提出了一种基于隐私保护的神经网络学习方法，该方法为神经网络梯度下降法引入隐私保护的功能，但是这种方法只适用于简单的神经网络模型，如没有隐藏层并且输出层只含有一个结点的网络模型。

文献[6]使用加密技术为前馈神经网络引入隐私保护的功能，从而在非线性分类的过程中，保证隐私信息不被泄露。该方法假定学习过程中只有一个神经网络模型拥有者，且该模型拥有者不拥有任何训练数据，而那些数据拥有者只能把各自的数据提供给神经网络，数据拥有者并不拥有模型。其目的是保证模型拥有者不会得到任何数据的信息，而数据拥有者也不会得到神经网络模型的任何信息。本文方法与文献[6]的不同之处在于：本文提出的算法是让所有参与方都安全地共享学习模型，每个参与方都拥有自己的数据，并且，学习过程中任何参与方都不需要将自己的数据泄露给其它参与方。

此外，文献[5]和文献[6]只是从理论上分析了各自提出的算法，并没有通过实验来验证所提算法的应用性和效率。本文将通过在现实世界数据集上进行实验，验证本文提出的算法具有良好的执行效率和隐私保护性能。

2 相关概念

本节首先介绍神经网络的基本概念以及神经网络的应用；然后介绍同态加密的概念和语义安全性的概念；最后介绍安全多方计算协议。由于这些安全多方计算协议都基于加密技术，因此可以保证在计算过程中，每个参与方的数据信息不会泄露给其他参与方。这些安全多方计算协议将作为本文算法的基础。

2.1 神经网络

神经网络是一种基于人脑的结构构造的信息处理系统，用于实现人脑的某些功能，是对人脑的一种简单的抽象或模仿[8]。神经网络常被用于解决一些复杂的问题，其最大特点就是自我学习功能，即通过对大量训练样本的反复学习，不断地对网络连接权值进行修改，从而使网络连接权值稳定分布在一个固定范围之内。神经网络能够对每个输入信号进行处理，确定其权值，然后确定所有输入信号的加权值，最后确定其输出，从而解决相关问题[9]。

神经元[10] 是神经网络的基本组成单元，设 $(x_1, x_2, \cdots, x_n)$ 为神经元的输入信号，$(w_1, w_2, \cdots, w_n)$ 为各个输入与神经元之间的连接权值。神经元接受来自外部的输入信号，将信号与各个边上的权值相乘并求和，即 $\sum_{j=1}^{n} w_j x_j$，将求得的加权减去阈值 θ，再将这个结果传递给函数 $f(u)$，得到最后的输出。其中函数 $f(u)$ 被称为激活函数，通常为 sigmoid 函数[11]，$f(u)=\frac{1}{1+e^{-u}}$。因此神经元的作用就是将来自外部的多个输入进行处理，得到相应的输出。

2.2 同态加密

同态加密是一种特殊的加密方法。该方法通过在密文中使用一种代数运算，从而允许在明文上进行特殊的代数运算。Paillier 在文献[12]中首次提出了同态加密方法。该方法可以被描述如下：

设 E 是加密函数，D 是解密函数。在生成密钥的过程中，随机选择两个素数 p 和 q，然后令 $n=p*q$，$\lambda=lcm(p-1, q-1)$。其中符号 lcm 表示计算最小公倍数。随机选择一个整数 $g\in Z^*_{n^2}$，公钥记为 (n, g)，私钥记为 (λ, μ)。其中 $\mu=(L(g^\lambda \bmod n^2))^{-1} \bmod n$，$L(u)=\frac{u-1}{n}$。

当 $u<n^2$ 并且 $u=1 \bmod n$ 时，对信息进行加密的步骤如下：

(1)随机选择一个 $r\in Z_n^*$；

(2)对信息 m 进行加密，得到 $E(m)=g^m * r^n \bmod n^2$。

对一条加密后的信息 c，进行解密的方法是：$m=D(c)=L(c^\lambda \bmod n^2) * \mu \bmod n$。

同态加密方法具有下列性质。

(1)允许在明文上运行同态的加法运算。该加法运算可以描述为：

$$D(E(m_1, r_1) * E(m_2, r_2) \bmod n^2)=m_1+m_2 \bmod n$$

其中，m_1 和 m_2 是任意两条明文信息，并且 r_1 和 r_2 分别是 m_1 和 m_2 对应的随机数。

(2)允许在明文上运行同态的乘法运算。该乘法运算可以描述为：

$$D(E(m_1, r_1)^{m_2} \bmod n^2)=m_1 * m_2 \bmod n$$

其中 m_1 和 m_2 是任意两条明文信息，并且 r_1 是 m_1 对应的随机数。

2.3 语义安全性

在研究公钥加密的安全性时，经常会使用到语义安全性

(semantic secutity)[13]的概念。语义安全性指恶意者在获得密文的情况下,不能有效地计算出比没有获得密文时更多的关于明文的信息,即恶意者通过使用密文,无法获得额外的有用信息。从语义安全性的定义来看,应该使用基于模拟器[14]的形式来为其下定义,即恶意者得到某个消息的密文,而模拟器却没有得到,此时恶意者能计算出关于该消息的所有信息,如果模拟器也能计算出,则说明加密是安全的。语义安全性经常被用来证明基于隐私保护的数据挖掘算法的安全性。文献[15]已经证明了同态加密具有语义安全性。

2.4　安全多方计算协议

通常情况下,数据挖掘和机器学习都希望数据被集中存放在一个地点,以便对数据进行分析。但是隐私保护要求各个参与方的数据不被集中存放。为了解决两者之间的矛盾,可以使用加密技术,设计安全多方计算协议[16],既可以实现在多个数据集合上联合执行某些计算,又可以保证各参与方的隐私数据不泄露给其他参与方。本文介绍的安全多方计算协议都基于加密操作。

(1)安全多方加协议

使用该协议,将私有输入值 x_i 的求和运算转化为私有输出值 y_i 的乘积运算,即:$\sum_{i=1}^{n} x_i = \prod_{i=1}^{n} y_i$。

安全多方加协议的具体步骤可描述如下:

(1)参与方 P_n 选择 $n-1$ 个数字 $x_{n,1}, x_{n,2}, \cdots, x_{n,n-1}$,使得 $x_n = x_{n,1} + x_{n,2} + \cdots + x_{n,n-1}$。

(2)每一个参与方 P_i,$1 \leqslant i \leqslant n-1$,和 P_n 在各自的输入值 x_i 和 $x_{n,i}$ 上运行安全双方加协议,使得 $x_i + x_{n,i} = y_{i,n} * y_n$,其中 $x_i \in P_i$,$x_{n,i} \in P_n$,$y_{i,n} \in P_i$,$y_n \in P_n$,$y_{i,n}$ 和 y_n 是参与方 P_i 和 P_n 各自的私有输出值。

因此可以得出:

$$\begin{aligned} x_1 + \cdots + x_n &= (y_{1,n} * y_n) + \cdots + (y_{n-1,n} * y_n) \\ &= (y_{1,n} + \cdots + y_{n-1,n}) * y_n \end{aligned}$$

(3)此时 $y_{1,n} + \cdots + y_{n-1,n}$ 是参与方 $P_1, P_2, \cdots, P_{n-1}$ 所具有的 $n-1$ 个元素的求和,因此协议将从步骤 1 重新开始执行,这个循环将被反复执行直到最终剩下两个参与方 P_1 和 P_2 进行求和计算后,循环将结束。此时就可以对 P_1 和 P_2 使用安全双方加运算协议。

(2)安全多方乘积协议

使用该协议将私有输入值 x_i 的乘积运算转化为私有输出值 y_i 的求和运算,即:$\prod_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$。

每个参与方 P_i 会使用到同态加密算法[17] E_i,并且具有公钥 e_i 和私钥 d_i。安全多方乘积协议的具体步骤可描述如下:

(1)参与方 P_1 使用同态加密算法 E_1 和参与方 P_2 在各自的私有输入值 x_1 和 x_2 上运行安全双方乘积协议,使得 $x_1 * x_2 = y_{1,1} + y_{2,1}$。所以,

$$\begin{aligned} x_1 * x_2 * x_3 \cdots * x_n &= (y_{1,1} + y_{2,1}) * x_3 * \cdots * x_n \\ &= (y_{1,1} * x_3 * \cdots * x_n) + (y_{2,1} * \\ &\quad x_3 * \cdots * x_n) \end{aligned}$$

(2)$(y_{1,1} * x_3 * \cdots * x_n)$ 和 $(y_{2,1} * x_3 * \cdots * x_n)$ 都表示 $n-1$ 个元素的乘积。并且 $(y_{1,1} * x_3 * \cdots * x_n)$ 作用在参与方 $P_1, P_3, \cdots, P_n$ 之间,$(y_{2,1} * x_3 * \cdots * x_n)$ 作用在参与方 $P_2, P_3, \cdots, P_n$ 之间。上面的步骤会被反复执行,直到求和的式子中的每一项都变成两个参与方的乘积形式才结束循环。此时可以在求和式子中的每一项上运行安全双方乘积协议。

3　反向传播神经网络学习算法

反向传播神经网络是一种多层前馈网络,通常由输入层、隐含层和输出层组成[18]。其中,每一层均由若干个结点组成,每一个结点代表一个神经元,隐含层中的神经元通常采用 Sigmoid 型激活函数来表示,而输入层或输出层的神经元,则通常采用线性传递函数来表示[19]。反向传播神经网络的上层结点与下层结点之间通过权值进行连接,同层内的结点之间没有联系。反向传播神经网络由正向传播和反向传播组成,在正向传播过程中,当大量样本输入神经网络后,输入信号通过输入层向前传播,经过隐含层,并由隐含层的函数作用后,将隐含层的输出信号传向输出层,并最终得到输出结果[20]。在整个正向传播过程中,每一层的神经元只接收来自上一层神经元的输入,而每一层的神经元的输出只会影响下一层的神经元的输出。如果最终的输出结果与期望结果之间存在较大误差,则转入反向传播过程,将误差值沿着原来的连接通道从输出层经过隐含层,最终回到输入层。它通过网络将误差信号沿原来的连接通路反传回来修改各层神经元的权值,直至达到期望目标[21]。文献[22]提出了一种反向传播神经网络学习算法,该算法的详细描述如下。

(1)设置步长 ρ,步长 ρ 的初始值应是一个较小的数;设置权向量 $\boldsymbol{W}$,权向量 $\boldsymbol{W}$ 的初始化值也应设置为较小值。

(2)选取一个训练样本数据 $\langle E^k, C^k \rangle$。

(3)正向传播阶段:从输入神经元开始,为每个神经元分别计算权向量和 S_i,利用激活函数计算 $u_i = f(S_i)$。

(4)反向传播阶段:从输出神经元开始,为每个神经元计算梯度:$f'(S_i) = u_i(1-u_i)$。如果 u_i 是输出单元,则 $\delta_i = (C_i - u_i) f'(S_i)$;对于其它单元,$\delta_i = (\sum_{m;m>i} w_{m,i}\delta_m) f'(S_i)$。

(5)更新权向量:$w_{i,j}^* = w_{i,j} + \rho\delta_i u_j$。

(6)如果终止条件被满足,则从建造好的神经网络中退出,否则将返回步骤 2,继续执行。

因为文献[22]提出的反向传播神经网络学习算法未考虑隐私保护的问题,比如输入数据中的隐私信息应该对每个参与方保密,学习模型中所含有的隐私信息也不能泄露给外界,所以本文对文献[22]的算法进行改进,增加了隐私保护的功能,并且所提出的算法适用于数据集合被水平分割的情况。

4　基于隐私保护的反向传播神经网络学习算法

针对数据集合被水平分割的情况,本文提出一种基于隐私保护的反向传播神经网络学习的算法 PPNN-DHP。

在数据被水平分割的情况下,每个参与方都拥有一些记录的所有属性值或拥有整个数据库的一些行记录。算法执行完之后,所有参与方可以安全地共享被建造好的学习模型,并且所有参与方可以使用该模型为各自的目标数据预测出相应的输出结果。

假设训练数据集合 D 被水平分割成 $D_1, D_2, \cdots, D_n$,数据集合 D 被分割后的这 n 个部分分别被参与方 $P_1, P_2, \cdots, P_n$ 所拥有。其中 $|D_i| = n_i$。

每个元素 $d_{i,j} \in D_i$,$1 \leqslant j \leqslant n_i$,每个元素 $d_{i,j}$ 表示为 $\langle \boldsymbol{E}_{i,j}, \boldsymbol{C}_{i,j} \rangle$,其中 $\boldsymbol{E}_{i,j} = \langle 1, u_{i,j,1}, u_{i,j,2}, \cdots, u_{i,j,p} \rangle$ 是输入向量,而 $\boldsymbol{C}_{i,j}$

是相对应的输出向量。该神经网络的权向量记为：

$$\boldsymbol{W}=\langle w_{p+1,0}\cdots,w_{p+1,p}\cdots,w_{p+k,0}\cdots,w_{p+k,p},w_{p+k+1,p+1}\cdots,w_{p+k+1,p+k}\rangle$$

神经网络学习的目的是为训练样本集合计算出网络权向量 $\boldsymbol{W}$。为了保护反向传播神经网络学习模型中的隐私信息不被泄露，同时为了保护权向量 $\boldsymbol{W}$ 的信息不被泄露，可以将权向量 $\boldsymbol{W}$ 分配给所有参与方，使得每个参与方都拥有权向量 $\boldsymbol{W}$ 的一部分私有值。设其中任一参与方 $P_i(1\leqslant i\leqslant n)$ 所拥有的权向量记为 $\boldsymbol{W}_i$。

$$\boldsymbol{W}_i=\langle w_{i,p+1,0}\cdots,w_{i,p+1,p}\cdots,w_{i,p+k,p},w_{i,p+k+1,p+1}\cdots,w_{i,p+k+1,p+k}\rangle$$

其中，权向量 $\boldsymbol{W}$ 中的每个元素可以通过式 $w_{p,s}=\sum_{i=1}^{n}w_{i,p,s}$ 来计算。

在算法的最开始，对权向量 $\boldsymbol{W}$ 进行初始化。

下面是本文提出的算法 PPNN-DHP 的具体步骤。

(1)从 n 个参与方中随机选择一个参与方 $P_i(1\leqslant i\leqslant n)$。

(2)P_i 随机生成一个整数 j，$(1\leqslant j\leqslant n_i)$，然后选择元素 $d_{i,j}$，而元素 $d_{i,j}$ 对应 $\langle\boldsymbol{E}_{i,j},\boldsymbol{C}_{i,j}\rangle$。

(3)正向传播阶段

1)隐含层中的神经元 $S_l,l\in\{p+1,\cdots,p+k\}$ 的计算如下：

$$S_l=\boldsymbol{E}_{i,j}\langle w_{l,0},\cdots,w_{l,p}\rangle=\boldsymbol{E}_{i,j}<w_{1,l,0},\cdots,w_{1,l,p}\rangle+\cdots+\boldsymbol{E}_{i,j}\langle w_{n,l,0},\cdots,w_{n,l,p}\rangle$$

在上式中，$\boldsymbol{E}_{i,j}\langle w_{i,l,0},w_{i,l,1},\cdots,w_{i,l,p}\rangle$ 由参与方 P_i 单独计算，因为运算符两边的运算量都属于这个参与方 P_i。

而 $\boldsymbol{E}_{i,j}\langle w_{m,l,0},w_{m,l,1},\cdots,w_{m,l,p}\rangle,m\neq i$，由参与方 P_i 和 P_m 使用安全点积协议计算。由于安全点积协议的最终结果被分配给每个参与方，从而使得安全点积协议的计算结果就等于对每个参与方的输出值进行求和。因此可以认为：$\boldsymbol{E}_{i,j}\langle w_{m,l,0},w_{m,l,1},\cdots,w_{m,l,p}\rangle=R_{l,i,m}+R_{l,m}$，其中 $R_{l,i,m}$ 和 $R_{l,m}$ 分别表示参与方 P_i 和 P_m 的输出值。从而进一步得出：

$$S_l=(R_{l,i,1}+\cdots+R_{l,i,n})+(R_{l,1}+\cdots+R_{l,i-1}+R_{l,i+1}+\cdots+R_{l,n})$$

其中，$R_{l,i,1},\cdots,R_{l,i,n}$ 属于参与方 P_i，而 $R_{l,m}(m\neq i)$ 属于参与方 P_m。如果假设 $R_{l,i,1}+\cdots+R_{l,i,n}=R_{l,i}$，则 $S_l=R_{l,1}+\cdots+R_{l,n}$。

2)通过使用 Sigmoid 函数计算隐含层的 $u_l,l\in\{p+1,\cdots,p+k\}$。

$$u_l=f(S_l)=f(R_{l,1}+\cdots+R_{l,n})=\frac{1}{1+\mathrm{e}^{-(R_{l,1}+\cdots+R_{l,n})}}=\frac{1}{1+\mathrm{e}^{-R_{l,1}}*\cdots*\mathrm{e}^{-R_{l,n}}}$$

使用安全多方乘积运算和安全多方加法运算，对上面的式子进行进一步的分析，得出：

$$\frac{1}{1+\mathrm{e}^{-R_{l,1}}*\cdots*\mathrm{e}^{-R_{l,n}}}=\frac{1}{1+(x_{1,l}+\cdots+x_{n,l})}=\frac{1}{y_{1,l}*\cdots*y_{n,l}}=y_{1,l}^{-1}*\cdots*y_{n,l}^{-1}=z_{1,l}*\cdots*z_{n,l}$$

令 $z_{m,l}=y_{m,l}^{-1}$，$1\leqslant m\leqslant n$，则 $u_l=\prod_{m=1}^{n}z_{m,l}$，$l\in\{p+1,\cdots,p+k\}$，并且 $z_{m,l}$ 被参与方 P_m 所拥有，$1\leqslant m\leqslant n$。

3)计算输出层的神经元 S_{p+k+1}。

$$\begin{aligned}S_{p+k+1}&=\langle 1,u_{p+1},\cdots,u_{p+k}\rangle\langle w_{p+k+1,0},w_{p+k+1,p+1},\cdots,w_{p+k+1,p+k}\rangle\\&=w_{p+k+1,0}+u_{p+1}*w_{p+k+1,p+1}+\cdots+u_{p+k}*w_{p+k+1,p+k}\\&=w_{1,p+k+1,0}+\cdots+w_{n,p+k+1,0}+z_{1,p+1}*\cdots*z_{n,p+1}*(w_{1,p+k+1,p+1}+\cdots+w_{n,p+k+1,p+1})+z_{1,p+2}*\cdots*z_{n,p+2}*(w_{1,p+k+1,p+2}+\cdots+w_{n,p+k+1,p+2})+\cdots+z_{1,p+k}*\cdots*z_{n,p+k}*(w_{1,p+k+1,p+k}+\cdots+w_{n,p+k+1,p+k})\end{aligned}$$

令 $A=z_{1,l}*\cdots*z_{n,l}*(w_{1,m,l}+\cdots+w_{n,m,l})$，使用安全多方乘积运算，得：

$$\begin{aligned}A&=z_{1,l}*\cdots*z_{n,l}*w_{1,m,l}+\cdots+z_{1,l}*\cdots*z_{n,l}*w_{n,m,l}\\&=(z_{1,l}*w_{1,m,l})*z_{2,l}*\cdots*z_{n,l}+\cdots+z_{1,l}*\cdots*z_{n-l,l}*(z_{n,l}*w_{n,m,l})\\&=z'_{1,l}*z_{2,l}*\cdots*z_{n,l}+\cdots+z_{1,l}*\cdots*z_{n-1,l}*z'_{n,l}\\&=(t_{1,l,1}+t_{2,l,1}+\cdots+t_{n,l,1})+\cdots+(t_{1,l,n}+t_{2,l,n}+\cdots+t_{n,l,n})\\&=(t_{1,l,1}+t_{1,l,2}+\cdots+t_{1,l,n})+\cdots+(t_{n,l,1}+t_{n,l,2}+\cdots+t_{n,l,n})\\&=r_1+r_2+\cdots+r_n\end{aligned}$$

其中，令 $z'_{j,l}=z_{j,l}*w_{j,m,l}$，$1\leqslant j\leqslant n$，$r_j=\sum_{h=1}^{n}t_{j,l,h}$。

所以 S_{p+k+1} 等于被涉及到的参与方的私有值的总和。

4)输出层计算神经网络的输出结果 u_{p+k+1}，模仿此算法步骤 3 中的 3)中计算 u_l 的方法，可以计算出 $u_{p+k+1}=\prod_{l=1}^{n}z_{l,p+k+1}$，其中 $z_{l,p+k+1}$ 是参与方 P_l 对应的私有值，$1\leqslant l\leqslant n$。

5 算法 PPNN-DHP 的安全性分析

定义 1(ε 差分隐私) 对于一种算法 PPDM，如果对于任意隐私数据 PD，都能够找出一个 ε，使其满足 $|Pr(PD|\text{PPDM})-P(PD)|\leqslant\varepsilon$，则此算法满足 ε 差分隐私[23]。

PD：表示隐私数据。

PPDM：表示基于隐私保护的算法。

PD_{P_i}：表示 P_i 的隐私数据。

EXT_{P_i}：表示 P_i 通过算法能够获得的额外的信息。

$GAIN_{P_i}$：表示 P_i 使用算法可以访问另一参与方的隐私数据的优势。

$GAIN_{SEC}$：表示 P_i 使用算法并通过阅读语义上的密文可以访问另一参与方的隐私数据的优势，这种优势在使用的同态加密系统中可以忽略不计。

$Pr(PD)$：不使用任何基于隐私保护的算法的情况下，隐私数据 PD 被泄露的概率。

$Pr(PD|\text{PPDM})$：使用基于隐私保护的算法 PPDM 后，隐私数据 PD 被泄露的概率。

$|Pr(PD|\text{PPDM})-Pr(PD)|$：使用隐私保护算法和不使用隐私保护算法情况下，隐私数据 PD 被泄露的概率值的差。

ε：隐私保护度，ε 的值越小，说明隐私保护程度越高。

定理 1 本文提出的 PPNN-DHP 算法满足 ε 差分隐私。

证明：根据定义 1，要证明算法 PPNN-DHP 满足 ε 差分隐私，只需要找出 ε，使其满足 $|Pr(PD|\text{PPNN-DHP})-Pr(PD)|\leqslant\varepsilon$。

因为每个参与方 P_j 使用该算法可以访问另一参与方的隐私数据的优势可以表示为：

$$GAIN_{Pj}=Pr(PD_{Pk}|EXT_{Pj},\text{PPNN-DHP})-Pr(PD_{Pk}|$$

EXT_{P_j}),$k\neq j$

当 $1\leqslant j\leqslant n$,$j\neq i$ 时,每个参与方 P_j 使用自己的随机生成向量只和参与方 P_i 运行安全点积协议,并且参与方 P_j 使用自己的私有输出值和其他参与方运行安全多方加协议,因此 $GAIN_{P_j}=GAIN_{\text{SEC}}$,$j\neq i$。

因为 $GAIN_{\text{SEC}}$ 表示 P_j 使用算法并阅读语义上的密文可以访问另一参与方的隐私数据的优势,这种优势在使用的同态加密系统中可以忽略不计。因此 $GAIN_{P_j}$ 可忽略不计。

参与方 P_i 对接收到的其他参与方的信息进行解密,并且对其他参与方的私有输出值的符号进行解密,其只会知道最终权向量中自己那一部分的私有值,而这一部分值实际上就是 P_i 的最终输出值。所以 P_i 无法通过算法获得额外的信息,也就无法预测其他参与方的私有数据。所以,可以认为:

$\varepsilon=\max(GAIN_{P_i},GAIN_{P_j})=GAIN_{P_i}$

因此,对于每一个 $k,j\in\{1,\cdots,n\}$,$k\neq j$,得出:

$Pr(PD_{Pk}\mid EXT_{P_j},\text{PPNN-DHP})-Pr(PD_{Pk}\mid EXT_{P_j})\leqslant GAIN_{P_i}=\varepsilon$

最终可以找到一个 $\varepsilon=GAIN_{P_i}$,使得 $|Pr(PD\mid\text{PPNN-DHP})-Pr(PD)|\leqslant\varepsilon$ 成立,所以 PPNN-DHP 算法满足 ε 差分隐私的性质。得证。

6　实验评价与分析

本部分将通过实验来表明本文算法的应用性和性能。整个实验由 Java 语言编程实现。实验使用的电脑配置如下:CPU 是 Intel Core i3,2.13GHz,内存是 4GB,硬盘是 500GB,操作系统是 Windows 7 Professional。

6.1　执行效率的评价与分析

本实验选取文献[23]提出的反向传播神经网络学习算法 NPPBNN 和本文提出的基于隐私保护的反向传播神经网络学习算法 PPNN-DHP 在执行时间上进行比较。算法 NPPBNN 不具备隐私保护功能。

实验结果如图 1 所示,在加密密钥长度和参与方数目不变的情况下,当计算结点数小于 13 时,基于隐私保护的反向传播神经网络学习算法 PPNN-DHP 所需的执行时间明显长于算法 NPPBNN。这是因为当计算结点数目较少时,算法 PPNN-DHP 在隐私保护子算法中消耗的时间较长。而当计算结点数大于 13 后,算法 PPNN-DHP 和算法 NPPBNN 的执行时间的差距将逐渐减小。这是因为当计算结点数较多时,PPNN-DHP 的隐私保护子算法中的安全多方计算会被分配到每个计算结点,进而缩短了隐私保护子算法所占用的时间。当计算结点数大于 256 之后,两种算法所需要的执行时间相当。

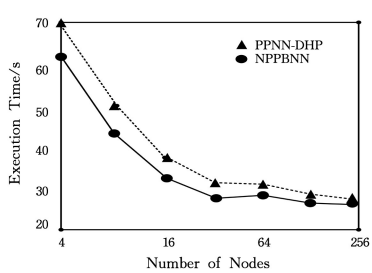

图 1　算法 PPNN-DHP 和 NPPBNN 在执行时间上的比较

Fig. 1　Comparison of execution time between algorithms PPNN-DHP and NPPBNN

6.2　准确度误差的评价与分析

通过下面的实验比较基于隐私保护的反向传播神经网络学习算法 PPNN-DHP 与非隐私保护的反向传播神经网络学习算法 NPPBNN 在准确度误差方面的差别。该实验使用的数据集来自于 UCI dataset repository。表 1 描述了数据集合和训练参数。该实验将在 Iris,Dermatalogy,Sonar,Landsat 数据集合上分别进行反向传播神经网络学习。针对每个数据集合的实验所需的测试样本都从该数据集合中被随机选取。对于 Iris 和 Sonar 数据集合,每次实验随机选取 20 个测试样本,而对于 Dermatalogy 和 Landsat 数据集合,每次实验随机选取 30 个测试样本。对于较大的数据集合,如 Landsat,则需要将训练回合数设置得较小。

为了保护每个参与方的数据隐私和中间计算结果不被泄露,可以在非隐私保护的反向传播神经网络学习算法中使用加密技术。而使用了加密操作,会产生准确度误差。针对每个数据集合的准确度误差可以通过下面的公式来计算。

准确度误差$=T_1-T_2$

其中,T_1 表示反向传播神经网络算法的隐私保护版本所对应的测试误差率,而 T_2 表示该算法的非隐私保护版本所对应的测试误差率。无论对于算法的隐私保护版本还是非隐私保护版本,测试误差率都可以通过下面的公式来计算。

$$\text{测试误差率}=\frac{\text{被误判的测试样本数}}{\text{训练样本总数}}$$

表 1　数据集合和训练参数

Table 1　Data set and training parameters

数据集合	样本总数	类数目	训练回合数
Iris	150	3	50
Dermatalogy	366	6	30
Sonar	208	2	60
Landsat	6435	6	10

在数据被水平分割的情况下,PPNN-DHP 和此算法的非隐私保护版本 NPPBNN 在测试误差率方面的实验结果如表 2 所列。

表 2　数据被水平分割情况下测试误差率比较

Table 2　Comparison of test error rate when the data is horizontally divided

(单位:%)

数据集合	NPPBNN	PPNN-DHP
Iris	12.00	16.87
Dermatalogy	19.20	25.03
Sonar	16.55	21.30
Landsat	14.48	17.35

通过观察表 2 中的实验结果可以发现,本文提出的算法的测试误差率高于非隐私保护的版本。原因是为了保护数据隐私而引入加密操作后,不可避免地出现了准确度误差。在数据被水平分割的情况下,4 个数据集合对应的准确度误差分别为 4.87%,5.83%,4.75%,2.87%。因为准确度误差在一定限度内,所以本文提出的基于隐私保护的反向传播神经网络学习算法 PPNN-DHP 对这些现实世界的数据集的学习仍然十分有效。

结束语　本文为反向传播神经网络提出基于隐私保护的算法,该算法适用于数据被水平分割的情况。并且该算法适用于多个参与方存在的分布式环境。所有的参与方在整个数据集合上联合建造一个神经网络学习模型,并且每个参与方

不需要将自己的数据透露给其他参与方。最后所有的参与方可以安全地共享学习模型，并且使用该模型为各自的目标数据预测出相应的输出结果。本文在实验部分表明了算法PPNN-DHP与非隐私保护版本算法NPPBNN在执行时间、准确度误差上的差别，并分析了原因。本文提出的基于隐私保护的反向传播神经网络学习算法仅适用于数据被水平分割情况，未来将讨论数据被垂直分割时，如何设计出基于隐私保护的反向传播神经网络学习算法。

参 考 文 献

[1] WANG J, WANG L. A new anonymity-based protocol preserving privacy based cloud environment [J]. Computer Modelling and New Technologies, 2014, 18(9): 139-144.

[2] WILAMOWSKI B M. Neural network architectures and learning algorithms [J]. Industrial Electronics Magazine, 2009, 3(4): 56-63.

[3] FRYE R C, RIETMAN E A, WONG C C. Back-propagation learning and nonidealities in analog neural network hardware [J]. IEEE Trans on Neural Networks, 1991, 2(1): 110-117.

[4] BARNI M, ORLANDI C, PIVA A. A privacy-preserving protocol for neural-network-based computation[C]// Proceedings of ACM MM&Sec'06. Geneva, Switzerland: ACM, 2006: 146-151.

[5] WAN L, NG W K, HAN S, et al. Privacy-preservation for gradient descent methods[C]// Proceedings of ACM SIGKDD' 07. New York: ACM, 2007: 775-783.

[6] SECRETAN J, GEORGIOPOULOS M, CASTRO J. A privacy preserving probabilistic neural network for horizontally partitioned databases[C]// Proceedings of IEEE IJCNN'07. Orlando, FL: IEEE, 2007: 1554-1559.

[7] CRAVEN M P, CURTIS K M, HAYES-GILL B R. Frequency division multiplexing in analogue neural network [J]. Electronics Letters, 1991, 27(11): 918-920.

[8] HUANG L, FENG D G, LIAN Y F, et al. Artificial-neural-network-based DDoS defense effectiveness evaluation [J]. Journal of Computer Research and Development, 2013, 50(10): 2100-2108.

[9] KODA M. Neural network learning based on stochastic sensitivity analysis [J]. IEEE Trans on Cybernetics, 1997, 27(1): 132-135.

[10] LI D L, LU D T, KONG X Y, et al. Implicit surfaces based on BP neural networks [J]. Journal of Computer Research and Development, 2007, 44(3): 467-472.

[11] ABHISHEK K, KUMAR A, RANJAN R, et al. A rainfall prediction model using artificial neural network[C]// Proceedings of IEEE ICSGRC'12. Shah Alam, Selangor: IEEE, 2012: 82-87.

[12] PASCAL P. Public-key cryptosystems based on composite degree residuosity classes [J]. Lecture Notes in Computer Science, 1999, 1592(1): 223-238.

[13] DHAKAR R S, GUPTA A K, SHARMA P. Modified RSA encryption algorithm [C]// Proceedings of IEEE ACCT' 12. Rohtak, Haryana: IEEE, 2012: 426-429.

[14] ORLANDI C, PIVA A, BARNI M. Oblivious neural network computing via homomorphic encryption [J]. Journal of Information Security, 2007, 20(7): 105-110.

[15] SAMET S, MIRI A. Privacy preserving ID3 using gini index over horizontally partitioned data [C]// Proceedings of IEEE AICCSA'08. Doha: IEEE, 2008: 645-651.

[16] CANETTI R. Security and composition of multiparty cryptographic protocols [J]. Journal of Cryptology, 2000, 13(1): 143-202.

[17] RANE S, SUN W, VETRO A. Secure function evaluation based on secret sharing and homomorphic encryption [C]// Proceedings of IEEE CCAC'09. Monticello, IL: IEEE, 2009: 827-834.

[18] LINDA O, VOLLMER T, MANIC M. Neural network based intrusion detection system for critical infrastructures [C]// Proceedings of IEEE IJCNN' 09. Atlanta, GA: IEEE, 2009: 1827-1834.

[19] RAFIQ M A, ROY N K, GHOSH B C. Three algorithms for learning artificial neural network: A comparison for induction motor flux estimation [C]// Proceedings of IEEE ICCIT '09. Dhaka: IEEE, 2009: 355-360.

[20] ADHIKARI S P, HYONGSUK K, BUDHATHOKI R K. A circuit-based learning architecture for multilayer neural networks with memristor bridge synapses [J]. IEEE Transaction on Circuits and Systems, 2015, 62(1): 215-223.

[21] SHEIKH S. Arabic-urdu script recognition through mouse: An implementation using artificial neural network[C]// Proceedings of IEEE ITNG'10. Las Vegas, NV: IEEE, 2010: 307-310.

[22] STEPHEN I, GALLANT. Neural network learning and expert systems [M]. Massachusetts: MIT Press, 1993.

[23] ZHAN J, MATWIN S. A crypto-based approach to privacy-preserving collaborative data mining[C]// Proceedings of IEEE ICDM'06. Hong Kong: IEEE, 2006: 546-550.

WANG Jian, born in 1981, Ph. D. His main research interests include privacy preserving and data protection.